ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/229049336

Cross-Domain Bootstrapping for Named Entity Recognition

Article · January 2011

CITATIONS
3

READS
136

2 authors, including:

Ralph Grishman
New York University
262 PUBLICATIONS **7,883** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Proteus Project View project

FUSE Terminology Extraction View project

All content following this page was uploaded by Ralph Grishman on 15 January 2015.

The user has requested enhancement of the downloaded file.

Cross-Domain Bootstrapping for Named Entity Recognition

Ang Sun Ralph Grishman
New York University
719 Broadway, Room 706
New York, NY 10003 USA
{asun, grishman}@cs.nyu.edu

ABSTRACT
We propose a general cross-domain bootstrapping algorithm for domain adaptation in the task of named entity recognition. We first generalize the lexical features of the source domain model with word clusters generated from a joint corpus. We then select target domain instances based on multiple criteria during the bootstrapping process. Without using annotated data from the target domain and without explicitly encoding any target-domain-specific knowledge, we were able to improve the source model's F-measure by 7 points on the target domain.

Categories and Subject Descriptors
I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text analysis*

General Terms
Languages

Keywords
named entity identification and classification, domain adaptation, bootstrapping

1. INTRODUCTION
Named Entity Recognition (NER) is a fundamental information extraction task with the objective of identifying and classifying proper names into certain pre-defined categories such as *persons*, *organizations* and *locations*. Supervised NER systems perform well when they are trained and tested on data from the same domain. However, when testing on a new domain which is different or even slightly different from the domain they were trained on, their performance usually degrades dramatically. For example, Ciaramita and Altun [8] reported that a system trained on the CoNLL 2003 Reuters dataset achieved an F-measure of 0.908 when it was tested on a similar Reuters corpus but only 0.643 on a Wall Street Journal dataset.

The performance degradation phenomenon occurs when one has access to labeled data in one domain (the *source domain*) but has no labeled data in another domain (the *target domain*). This is a typical situation as one might be able to expend the limited effort required to annotate a few *target* examples as a test bed but cannot afford to annotate additional examples for training purpose. However, it is usually the case that we have access to abundant unlabeled data in the target domain.

This paper works on this common scenario where we have access to labeled data in the *source domain* and only unlabeled data in the *target domain*. We propose a cross-domain bootstrapping (CDB) algorithm to iteratively adapt the source domain model to the target domain. Specifically, we first train an MEMM (maximum entropy Markov model [27]) source/seed model using the labeled data in the source domain and then apply it to the unlabeled data pool of the target domain. We then select *good* instances based on multiple criteria and use them to re-train and upgrade the seed model.

CDB differs from previous bootstrapping algorithms in several aspects. First, the seed model is generalized with word clusters. A model trained on the source domain may perform poorly on the target domain partly because there are many target domain specific words (for both names and context words) that have not been observed in the source domain. This motivates our work to use word clusters as additional features to generalize the seed model. The assumption is that even if we have not observed a *target* word *Wt* in the source domain, another word *Ws* in the source domain might share the same cluster membership with the word *Wt*. The cluster level feature still fires even if the lexical feature is absent from the source domain. More specifically, we mix the labeled source domain corpus with the unlabeled target domain corpus and generate the Brown word clusters (Brown et al., [5]) from this joint corpus. We then extract cluster memberships as features to augment the feature based NER system trained on the source domain.

CDB is novel in its multi-criteria-based instance selection method. Standard bootstrapping usually adopts a single criterion which is based on the *confidence* measure only, promoting those instances that are most confidently labeled from the unlabeled data. This might not be a problem when the data used for training the seed model and the unlabeled data are drawn from the same domain. However, in our cross domain setting, the most confidently labeled examples are those that have been observed in or are most similar to the source domain. CDB uses multiple criteria to select instances that are novel, confident, representative and diverse. It first uses novelty as a filter, maintaining only these instances that are specific to the target domain. It then ranks these novel instances based on a confidence measure. Top ranked instances contribute to a candidate set. Finally, it applies representativeness

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright is held by the author/owner(s).
EOS, SIGIR 2011 workshop, July 28, Beijing, China.

and diversity measures to all the candidates and selects a subset of them for promotion.

The rest of this paper is organized as follows: The next section positions us with respect to related work. Section 3 briefly introduces our NER task and source and target domains. Section 4 describes the CDB algorithm in detail. We present an experimental study in Section 5 and conclude in Section 6.

2. Related Work

There is a large body of domain adaptation research on different NLP tasks. Due to space limitations, we only discuss work related to NER.

Supervised domain adaptation for NER works on the scenario where one has labeled data from both the source and the target domains [11, 14]. Daumé III [11] has shown that a better model can be learned from the labeled data by making three copies of the features: general, source-dependent and target-dependent. Without labeled data from the target domain, it is impossible to distinguish and jointly learn the three types of features. Our work also generalizes and augments features but is obviously different from the above approaches in that the word cluster features are extracted from an unlabeled corpus.

Semi-supervised domain adaptation for NER deals with the situation such as ours where one only has labeled data from the source domain but not the target domain [23, 39]. (We can also refer to this branch of research as unsupervised learning because there is no supervision from the target domain.) Jiang and Zhai [23] studied domain adaptation from an instance weighting perspective and proposed a balanced bootstrapping algorithm in which the small number of instances promoted from the target domain was re-weighted to have an equal weight to the large number of source instances. Their instance selection was based on a confidence measure. Wu et al. [39] described a domain adaptive bootstrapping framework where the instances were selected based on informativeness. Neither of the two approaches generalized their seed models as we have done and both of them used a single instance selection criterion instead of the multiple criteria we have used.

Standard bootstrapping for domain-specific NER or semantic lexicon acquisition works on the *target domain* directly (both the seed examples and the unlabeled data are from the target domain) and typically adopts a confidence measure for selecting new instances [20, 28, 31, 40]. It has been shown that seed selection is very important for standard bootstrapping [37]. The way we generalize our seed model is similar, but not identical to seed selection in a sense that both of the approaches try to provide a better starting point for bootstrapping.

3. Task and Domains

Our NER task is similar to those defined in some benchmark evaluations such as MUC-6 [18], CoNLL-2003 [35] and ACE-05 [1]. Given a raw sentence, the goal is to identify name expressions and classify them into one of the following three types: PER (person), ORG (organization) and GPE (Geo-Political entity). We choose to work with these three types as they are the most frequent ones in our target domain. Figure 1 illustrates examples from both domains.

Source Example: <GPE>U.S.</GPE> <ORG>Defense</ORG> *Secretary* <PER>Donald H. Rumsfeld</PER> *discussed the resolution …*

Target Example1: *The ruler of* <GPE>Saudi Arabia</GPE> *is* <PER>Fahad bin Abdul Aziz bin Abdul Rahman Al-Sa ?ud</PER>.

Target Example2: … *where Sheikh* <PER>Abdul Sattar al-Rishawi</PER> *and the* <ORG>Anbar Salvation Front</ORG> *became a force for stability.*

Figure 1: Examples of NER task and domains

Our **target domain** documents are from publicly available reports (in English) on terrorism, such as those from the Combating Terrorism Center at West Point[2]. There are many domain-specific characteristics of our target domain. Just to mention a few: it is noisy as we automatically convert PDF documents to text files (footnotes might be inserted in the middle of natural language sentences and punctuation might not be converted correctly such as the example "Al-Sa?ud"); it is Arabic name (transliterated) rich and the naming convention is different from English names; name variation is another noticeable problem.

We choose the ACE-05 annotated data as the **source domain** because the degree of overlap between the ACE-05 data and the target domain data is higher than the MUC-6 and the CoNLL-2003 datasets, which are from the 90s.

4. Cross-Domain Bootstrapping

We first present an overview of the CDB algorithm, and then we describe in detail the generalization of a seed model with word clusters and the multi-criteria-based instance selection method.

4.1 Overview of the CDB Algorithm

The input to the algorithm is a labeled dataset from the source domain and an unlabeled dataset from the target domain, denoted by D_S^L and D_T^U respectively. Let G denote the growing set which contains selected instances during each round t and is initialized to be an empty set at round 0; the CDB algorithm repeats the following steps until it meets a stopping criterion.

1. Train an NER seed model M_t with $D_S^L \bigcup G_t$, *generalize* it with word clusters
2. Label D_T^U using M_t
3. Select $D_T^L \subseteq D_T^U$ based on *multiple criteria*
4. Update: $G_{t+1} = G_t \bigcup D_T^L$ and $D_T^U = D_T^U \setminus D_T^L$

The output of the CDB algorithm is an NER model which will be used to identify and classify named entities in the target domain. It is important to mention that the seed model M is generalized with word clusters at each round, not just at the beginning of the algorithm.

[1] http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v3.pdf

[2] http://www.ctc.usma.edu/publications/sentinel

4.2 Seed Model Generalization

Ungeneralized seed model: NER is typically viewed as a sequential prediction problem. Given a sentence containing a sequence of tokens, the goal is to assign a name class to each one of the tokens. Formally, let $s = (t_1, ..., t_N)$ be an input sequence of tokens and $c = (c_1, ..., c_N)$ be the output sequence of name classes, the prediction problem is to estimate the probability $P(c \mid s)$. To facilitate the learning procedure, we use the standard BIO decoding scheme. Each name type *c*, other than the type *O* (not a name), is split into subtypes *B-c* (beginning of *c*) and *I-c* (continuation of *c*). Although the less used BILOU (beginning, inside, last, outside and unit-length) scheme was claimed to outperform the BIO scheme in [30], we did not observe the same behavior in our target domain (see Section 5.2). The BIO representation gives us 7 classes (3 name types $\times$ 2 subtypes + 1 not a name class).

We build an MEMM [27] model with the following customary features: 1) current token t_i; 2) lower case tokens in a 5-token-window $(t_{i-2}, t_{i-1}, t_i, t_{i+1}, t_{i+2})$; 3) a word type feature (all-capitalized, initial-capitalized, all-digits, etc.) for each token in the context window; 4) previous prediction c_{i-1}; 5) conjunction of c_{i-1}, 1), 2) and 3); 6) gazetteer membership of context tokens in a dictionary of country and US state names. Note that we do not extract POS tags as features since a good target domain POS tagger is not available to us.

As the model makes a local decision, that is, it predicts the name class for each individual token based on its feature vector, it might produce illegal transitions between name classes (e.g., B-PER followed by I-ORG). So we run the Viterbi algorithm to select the sequence of name classes with the highest probability.

Generalizing seed model with word clusters: The sparsity of lexical features is a notorious problem in many supervised NLP systems. Recent advances in generating word classes from unlabeled corpora and adding them as features has proven to be an effective way of generalizing the lexical features to alleviate sparsity [29, 30, 36]. The unavailability of a cross-domain unlabeled corpus hinders the direct adaptation of this technique to our cross-domain setting. Ideally, we would prefer an unlabeled corpus containing words of both domains so that the word classes can generalize for both domains. So we propose to generate a joint corpus by mixing the labeled source data with the unlabeled target data.

We then follow previous research and use the Brown algorithm [5] to generate word clusters from the joint corpus. The Brown algorithm is a hierarchical clustering algorithm which initially assigns each word to its own cluster and then repeatedly merges the two clusters which cause the least loss in average mutual information between adjacent clusters based on bigram statistics. By tracing the pairwise merging steps, one can obtain a word hierarchy which can be represented as a binary tree. A word can be compactly represented as a bit string by following the path from the root to itself in the tree, assigning a 0 for each left branch, and a 1 for each right branch. Table 1 shows some words and their bit string representations obtained from the joint corpus.

By using prefixes of different lengths one can produce word clusters of various granularities so as to avoid the commitment to a single cluster. We used clusters with lengths 4, 6, 10 and 20 and augmented the *previous*, *current* and *next* token features with word clusters [29, 30, 36]. For example, when we extract features for the *current* token "John", we will add a cluster feature *curPrefix6 = 110100* when we use length 6. (Note that the cluster feature is a nominal feature, not to be confused with an integer feature.) Now, even if we have not observed "Abdul" in our source domain, its cluster level feature still fires given that the *curPrefix6* feature is the same for both "John" and "Abdul".

Table 1: An example of words and their bit string representations. Bold ones are transliterated Arabic words.

Bit string	Examples
110100011	*John, James, Mike, Steven, Dan, …*
11010011101	***Abdul, Mustafa, Abi, Abdel**, …*
11010011111	***Shaikh, Shaykh, Sheikh, Sheik**, …*
1101000011	*President, Pope, Vice, …*
111111110	***Qaeda, Qaida, qaeda, QAEDA**, …*
00011110000	*FDA, NYPD, FBI, …*
000111100100	***Taliban**, …*

It is worth mentioning an additional benefit of using word clusters: different Arabic name variants are grouped together such as variants of "Shaikh" and variants of "Qaeda" in Table 1. Without analyzing and comparing the internal structure of the words, such as computing the edit distance between different words, the clustering algorithm itself is able to capture this domain-specific knowledge.

4.3 Multi-Criteria-based Instance Selection

Most standard bootstrapping algorithms use a confidence measure as the single selection criterion. In practice, this works well under the single domain setting. In a cross-domain setting like ours, the most confidently labeled instances are highly correlated with the source domain and hence contain little information about the target domain. In contrast, the CDB algorithm adopts an instance selection method based on multiple criteria.

Instance: We define an instance $I = \langle v, c \rangle$ as the feature vector *v* and the name class *c* of the current token t_i under consideration. Although *sentence* seems to be a more natural unit than *token* for a bootstrapped NER system [21], our sentences contain many target domain specific names and context words which undermines the reliability of any confidence measure defined over a sentence. However, when broken down to the token level, it is easy to design a relatively reliable confidence measure, as the feature vector *v* is essentially extracted from a short context window $(t_{i-2}, t_{i-1}, t_i, t_{i+1}, t_{i+2})$. Also, the feature vector does contain the transition from the previous name class to the current class as we include the prediction of the previous token t_{i-1} as a feature (The class of the previous token is known after we run Viterbi over the whole sentence). Moreover, the NER model outputs normalized probabilities predicting the name classes based on the vector *v* and it is convenient to add the vector to the feature file for re-training the NER model.

Novelty: Novelty prefers an instance *I* that contains target-domain-specific tokens in its context

window $(t_{i-2}, t_{i-1}, t_i, t_{i+1}, t_{i+2})$ and can be confidently labeled by the seed model. If all the context tokens have been observed in the source domain then the instance contains less target domain information than others. However, if all the 5 tokens are target-domain-dependent then the seed model's prediction of the instance might not be reliable. So we tried different values (1, 2 and 3) for the number of target-domain-specific tokens in the context window and different positions (token index in the range [*i-2, i+2*]) and found that the following simple measure worked the best: if the current token t_i is the only target-domain-specific token then the instance is considered to be novel.

Confidence: A reliable confidence measure is crucial to the success of a bootstrapping algorithm. If one bogus instance is selected, it will leads to the selection of many other bogus instances. CDB's confidence measure not only considers how confident an instance is labeled locally but also globally.

The ***local confidence*** of an instance *I* is defined as the posterior entropy of the 7 name classes C given the instance's feature vector *v*.

$$LocalConf(I) = -\sum_{c_i} p(c_i \mid v) \log p(c_i \mid v)$$

It is non-negative. It achieves its minimum value 0 when the MEMM model predicts a class c_i with probability 1 (more precisely when $p(c_i \mid v) = 1$). It achieves its maximum value when the predictions are evenly distributed over the 7 classes. So the lower the value, the more confident the instance is.

The ***global confidence*** concerns how other occurrences of the *current* token t_i in the instance *I* are labeled in the whole corpus. The linguistic intuition here is that one name usually belongs to one class in a corpus [13, 24, 38]. The CDB algorithm would prefer to select name tokens that can be consistently labeled in the whole corpus. So we gather all other occurrences of t_i in the corpus, generate their feature vectors and take as the name class of each occurrence the one with the highest probability returned by the MEMM model. The BI tags of the class are then deleted, for example, B-PER would become PER. This is because a name token can be at different positions (e.g., *Smith* can be either B-PER or I-PER). So global confidence uses 4 name classes instead of 7. We then compute the global confidence as below:

$$GlobalConf(I) = -\sum_{c_i} p(c_i) \log p(c_i)$$

where $p(c_i)$ is the corpus level probability of t_i belonging to the class c_i. It is defined as the number of times t_i is predicted with the class c_i divided by the total number of occurrences of t_i in the corpus. For example, if "Abdul" appears 10 times in the corpus and is predicted 8 times as PER, then the probability of "Abdul" belonging to the class PER is 0.8. Similar to the *local confidence* measure, the lower the value of the *global confidence*, the more confident the instance is.

We then propose a final measure to combine the two confidence measures. We simply take the product of the two measures.

$$ComConf(I) = LocalConf(I) \times GlobalConf(I)$$

Density: In addition to the most confident instances, CDB also aims to select the most representative instances. We use a density measure to evaluate the representativeness of an instance. The density of an instance *i* is defined as the average similarity between *i* and all other instances *j* in the corpus. The most representative instance is the one with the largest density value.

$$Density(i) = \frac{\sum_{j=1 \wedge j \neq i}^{N} Sim(i, j)}{N - 1}$$

where *N* is the total number of instances in the corpus and $Sim(i, j)$ is the similarity between the two instances, which is defined as the standard *Jaccard Similarity* between the feature vectors *u* and *v* of the two instances. The *Jaccard Similarity* between *u* and *v* is defined as the number of matched features of *u* and *v* divided by the number of unique features in the union of *u* and *v*. The match function for a feature *f* returns 1 if the values of *f* in *u* and *v* are the same and 0 otherwise.

$$Sim(u, v) = \frac{|u \bigcap v|}{|u \bigcup v|}$$

Alternatively, we could find the angle between the two feature vectors and compute the *Cosine Similarity* between them. However, as all the features for NER take discrete values the simpler *Jaccard Similarity* suffices to capture the similarity between feature vectors.

Diversity: CDB also aims to select instances as diverse as possible. Intuitively, if we have observed an instance and its similar instances a sufficient number of times then we cannot learn more *new* information from them. Take the instance "*, said * in his*" for example, where * is the *current* token, which we restrict to be a target-domain-specific token (*novelty*) and is highly likely to be a *person*; it is *confident* at both local and global levels given that the context is salient and it is probably very *dense* too. However, repeatedly selecting such instances is a waste of time because no additional benefit can be gained for CDB.

So *globally*, once an instance has been selected, it is removed from the unlabeled target corpus. The CDB algorithm will never select it again in the following rounds. *Locally*, in a single round, when we evaluate an instance *i*, we will compare the *difference* between *i* and all the instances *j* that have already been selected. If the *difference* is large enough, we accept *i*; otherwise we reject it. One possibility of measuring the *difference* is to directly use the similarity measure $Sim(i, j)$. But this tends to reduce the chance of selecting *dense* instances given that a dense instance has many similar instances and tends to occur more frequently than others. For example, if we already selected the instance "*, said Abdul in his*", the chance of selecting other similar instances "*, said * in his*" is low. We then turn to a compromise measure to compute the *difference* between instances *i* and *j* which is defined as the difference of their density values. By setting a small threshold for $diff(i, j)$, dense instances still have a higher chance to be selected while a certain degree of diversity is achieved at the same time.

$$diff(i, j) = Density(i) - Density(j)$$

Order of applying different criteria: CDB first applies the *novelty* measure to all the instances in the corpus to filter out non-novel instances, and then it computes the *confidence* score for each novel instance. Instances are then ranked in increasing order of confidence score (lower value means higher confidence) and the top ranked *M* instances will be used to generate a candidate set. CDB now applies the *density* measure to all the members in the candidate set and ranks the instances in descending order of density (larger value means higher density). Finally, CDB accepts the first instance (with the highest density) in the candidate set and selects other candidates based on the *diff* measure.

5. Experiments

5.1 Data, Evaluation and Parameters

Source domain data: Table 2 summarizes the source domain data (ACE 2005) used in this paper. The 2005 dataset contains 6 genres: Broadcast Conversations (bc), Broadcast News (bn), Conversational Telephone Speech (cts), Newswire (nw), Usenet (un) and Weblog (wl). We randomly selected 10 documents from each genre for testing purposes.

Table 2: Source domain data.

Genre	Training (#doc)	Test (#doc)
bc	50	10
bn	216	10
cts	29	10
nw	97	10
un	39	10
wl	109	10
Total	540 (285K words)	60 (31K words)

Target domain data: Table 3 lists the sizes of the unlabeled and the labeled corpus as well as the number of instances of the 3 name types in the labeled corpus. The labeled data (for testing purpose) is annotated according to the ACE 2005 guideline[3]. This test data and the word clusters generated from the TDT5 are available at http://cs.nyu.edu/~asun/#DataSet&Software.

Table 3: Target domain data.

Data	Size
Unlabeled/Labeled	10M/23K words
PERSON	771 instances
ORGANIZATION	585 instances
GPE	559 instances

Corpora for generating word clusters: To study the impact of unlabeled corpora on cross-domain NER, we downloaded the word clusters generated by Ratinov and Roth [30] and Turian et al., 2010 [36]. Following them, we used Liang's implementation of the Brown algorithm and generated 1,000 word clusters for both the TDT5 (the English portion) and the joint corpus [25]. The TDT5 is selected because it contains news from the year 2003 and some ACE 2005 training documents are also from 2003.

Table 4: Corpora for generating word clusters

Data	Size(#words)
Reuters 1996[4]	43M
Cleaned RCV1 [5]	37M
TDT5 (LDC2006T18)	83M
Joint Corpus (ACE training + Unlabeled Target data)	10M

Evaluation: Evaluation is done at the named entity level, not the BIO tags level. Boundary errors are penalized. We used the standard precision, recall and F1 score.

Parameters: As the CDB algorithm uses several parameters, we summarize them in Table 5 for easy reference. Because CDB runs the Viterbi algorithm on each sentence, it is time consuming to run it on the whole unlabeled data. So we divided them sequentially into 6 batch sets and picked a random set for bootstrapping in each iteration. The candidate set contains more instances than the CDB algorithm will actually select because of the density and diversity measures. As the majority of tokens belong to the not-a-name class, we select the same amount of name/not-a-name instances in order to provide a balanced distribution between the name and the not-a-name classes. We tried many different parameter values and those in Table 5 were selected by eyeballing the quality of selected instances.

Table 5: Parameters for CDB.

Parameter	Size or Value
Batch Set	60K sentences (roughly 1.7M tokens)
Candidate Set	2000/2000 name/not-a-name instances
D_T^L	300/300 name/not-a-name instances
Iterations	30
$diff(i, j)$	0.001

5.2 Performance of the Source Domain Model

We build two *source* models using the ACE 2005 training data in Table 2. The first model is an HMM model with the BILOU states encoding. The second model is the MEMM model with the BIO encoding using the conventional features as described in Section 4.2 (no word cluster features). Their performances on both the source and the target domains are summarized in Table 6.

Table 6 shows that although both models achieve F1 of 80s on the source domain, they generalize poorly on the target domain. Comparing the HMM model with the MEMM model, it seems that the feature based model generalizes better to the target domain than the generative HMM model even though we did use the word type features as a back-off model similar to [1]. We did

[3] http://projects.ldc.upenn.edu/ace/docs/English-Entities-Guidelines_v5.6.1.pdf

[4] http://cogcomp.cs.illinois.edu/page/software_view/4

[5] http://metaoptimize.com/projects/wordreprs/

not observe the advantage of using the BILOU scheme as reported in [30] for both the source and the target domains. Although we could not determine the exact reason why this happens for the source domain, for the target domain, it contains long transliterated Arabic names implying that the state transition from "I" to "I" is more common in the target than the source domain. This type of transition might not be observed enough and estimated sufficiently by the fine grained BILOU scheme.

Table 6: Performance of source models over the 3 name types

Model	P	R	F1	Domain
HMM(BILOU)	82.49	81.16	81.82	Source
HMM(BILOU)	53.29	57.52	55.33	Target
MEMM(BIO)	84.68	81.54	83.08	Source
MEMM(BIO)	70.02	61.86	65.69	Target

5.3 Performance of the Generalized Model

We augmented the source model with word clusters (as described in Section 4.2) from the four unlabeled corpora in Table 4. Their performance on the target domain is shown in Table 7.

Table 7 shows the superiority of using a joint corpus to generate word clusters: the 10M words joint corpus outperformed the other 3 larger corpora. The TDT5 corpus is more than 8 times larger than the joint corpus, but is still 1 point behind. Using word clusters from the Reuters corpora (Reuters 1996 and RCV1) have shown to improve NER systems' performance on the CoNLL 2003 NER task [30, 36]. But they provided limited performance gain for our model when testing on the target domain. The results shown here indicate the necessity of using a joint corpus or ideally a general purpose corpus for generalizing the source domain model for cross-domain NER.

Table 7: Performance of the generalized source model

Word clusters	P	R	F1
No Cluster	70.02	61.86	65.69
Reuters 1996	69.26	64.26	66.67
RCV1	66.33	64.42	65.36
TDT5	70.76	66.51	68.57
Joint Corpus	72.82	66.61	**69.58**

5.4 Performance of CDB

We start with the generalized source/seed model and run the CDB algorithm on the unlabeled target domain corpus using the parameters specified in Table 5. As mentioned earlier, the seed model is generalized sequentially, that is, word cluster features are used during each round. We plot F1 score against iteration in Figure 2. The results are obtained by testing the updated model during each round on the labeled target domain data. The results are averaged on 10 runs.

There are several clear trends in Figure 2. First, without using the novelty measure (the line at the bottom), CDB performs worse than GSM. Although the seed model is already generalized with word clusters, the most confidently labeled instances might still be more similar to the source than the target domain. This indicates that novelty is a necessary measure for cross-domain NER. Comparing the two confidence measures: *ComConf* and *LocalConf*, in general, *ComConf* outperforms *LocalConf*. After using the novelty measure, all instances are *new* to our seed model. So there is some degree of uncertainty when the model tries to make a local prediction. Not only considering the local prediction, but also considering how the same token is labeled globally, the *ComConf* measure seems to be a better choice in a cross-domain setting.

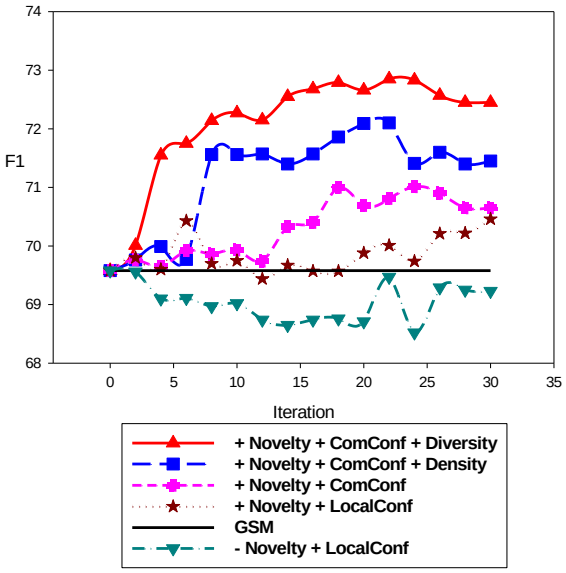

Figure 2: Performance of CDB. GSM stands for *generalized seed model* at iteration 0; + means with and – means without.

Regarding the density and the diversity measures, both of them further improve the performance. Density, however, does not perform well in the first 6 iterations. We checked the instances that had been selected during these iterations and found that many of them appear with very *strong* context words such as *Mr.*, *President*, *General* and *said*. They are considered representative instances according to our density measure. They can be regarded as cross-domain contexts which might have been learned by the generalized and un-generalized source domain models. In contrast, the diversity measure not only considers how representative an instance is but also prefers a certain degree of difference among the selected instances. Hence, the diversity measure has achieved the best result CDB could get so far. The best F score with the diversity measure is 72.85, a 7.16 improvement compared to the source model. The F score at the last iteration is 72.45, a 6.76 improvement compared to the source model.

We also run a standard bootstrapping procedure with the un-generalized seed model and with the same parameters used for the CDB procedure. The performance trends of using different instance selection criteria are similar to those of the CDB algorithm. The best F score, 70.12, is also obtained with the diversity measure. This further confirms that the multiple criteria proposed in this paper are better than a single criterion. CDB with generalized seed model outperformed the standard bootstrapping by more than 2 points which further indicates the usefulness of the combination of feature generalization and multi-criteria-based instance selection methods proposed in this paper.

6. Conclusion

We have described a general cross-domain bootstrapping algorithm for adapting a model trained only on a source domain to a target domain. We have improved the source model's F score by around 7 points. This is achieved without using any annotated data from the target domain and without explicitly encoding any target-domain-specific knowledge into our system. The improvement is largely due to the feature generalization of the source model with word clusters and the multi-criteria-based instance selection method.

Our immediate future work is to find a natural stopping criterion for the bootstrapping procedure, perhaps through the detection of *semantic drift* [28, 34]. Gazetteer resources have proven to be a powerful knowledge base for improving NER performance [9]. The only gazetteer in CDB now is a country and US state list. So another promising research avenue is to study how to automatically learn or mine a target domain named entity dictionary to further improve our system's performance.

7. ACKNOWLEDGMENTS

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL) contract number FA8650-10-C-7058. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

8. REFERENCES

[1] D. M. Bikel, S. Miller, R. Schwartz and R. Weischedel. 1997. Nymble: a high performance learning name-finder. In *Proc. of ANLP*.

[2] J. Blitzer, R. McDonald, and F. Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. of EMNLP*.

[3] A. Borthwick, J. Sterling, E. Agichtein and R. Grishman. 1998. Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proc. of Sixth WVLC*.

[4] A. Borthwick. 1999. A Maximum Entropy Approach to Named Entity Recognition. Ph.D. thesis, New York University.

[5] P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

[6] C. Chelba and A. Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help a lot. *In Proc. of EMNLP*, pages 285–292.

[7] H. L. Chieu and H. T. Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of CoNLL-2003*.

[8] M. Ciaramita and Y. Altun. 2005. Named-entity recognition in novel domains with external lexical knowledge. In *Advances in Structured Learning for Text and Speech Processing Workshop*.

[9] W. W. Cohen and S. Sarawagi. 2004. Exploiting Dictionaries in Named Entity Extraction: Combining SemiMarkov Extraction Processes and Data Integration Methods. In *Proc. of KDD*.

[10] H. Daumé III and D. Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

[11] H. Daumé III. 2007. Frustratingly easy domain adaptation. In *Proc. of ACL 2007*.

[12] M. Dredze, J. Blitzer, P. Talukdar, K. Ganchev, J. Graca, and F. Pereira. 2007. Frustratingly Hard Domain Adaptation for Parsing. In *Proc. of CoNLL*.

[13] J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL 2005*.

[14] J. R. Finkel and C. D. Manning. 2009. Hierarchical bayesian domain adaptation. In *Proc. of NAACL*.

[15] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proc. of HLT-NAACL*.

[16] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of CoNLL-2003.*

[17] R. Florian; J. Pitrelli; S. Roukos; I. Zitouni. Improving Mention Detection Robustness to Noisy Input. In *Proc. of ENMLP 2010*.

[18] R. Grishman and B. Sundheim. Message Understanding Conference - 6: A Brief History. In *Proceedings of the COLING, 1996*.

[19] R. Grishman, D. Westbrook and A. Meyers. 2005. NYU's English ACE 2005 System Description. In *Proc. of ACE 2005 Evaluation Workshop*. Washington, US.

[20] R. Huang and E. Riloff. 2010. Inducing Domain-specific Semantic Class Taggers from (Almost) Nothing. In *Proc. of ACL*.

[21] H. Ji and R. Grishman. 2006. Data Selection in Semi-supervised Learning for Name Tagging. In *ACL 2006 Workshop on Information Extraction Beyond the Document*.

[22] J. Jiang and C. Zhai. 2006. Exploiting domain structure for named entity recognition. In *Proceedings of HLT-NAACL'06*.

[23] J. Jiang and C. Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proceedings of ACL*.

[24] V. Krishnan and C. D. Manning. 2006. An effective two stage model for exploiting non-local dependencies in named entity recognition. In *Proc. of ACL*.

[25] P. Liang. 2005. Semi-Supervised Learning for Natural Language. Master's thesis, Massachusetts Institute of Technology.

[26] D. Lin and X. Wu. 2009. Phrase Clustering for Discriminative Learning. In *Proceedings of the ACL and IJCNLP 2009*.

[27] A. McCallum, D. Freitag and F. Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proc. of ICML*.

[28] T. McIntosh. 2010. Unsupervised discovery of negative categories in lexicon bootstrapping, In *Proc of EMNLP*.

[29] S. Miller, J. Guinness and A. Zamanian. 2004. Name Tagging with Word Clusters and Discriminative Training. In *Proc. of HLT-NAACL*.

[30] L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL-09*.

[31] E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of AAAI-99*.

[32] B. Roark and M. Bacchiani. 2003. Supervised and unsupervised PCFG adaptation to novel domains. In *Proc. of HLT-NAACL*, pages 126–133.

[33] D. Shen, J. Zhang, J. Su, G. Zhou, and C. Tan. 2004. Multi-criteria-based active learning for named entity recognition. In *Proceedings of ACL*.

[34] A. Sun and R. Grishman. 2010. Semi-supervised Semantic Pattern Discovery with Guidance from Unsupervised Pattern Clusters. *In Proc. of COLING*.

[35] E. Tjong and F. D. Meulder. 2003. Introduction to the conll-2003 shared task: Language independent named entity recognition. In *Proceedings of Conference on Computational Natural Language Learning*.

[36] J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*.

[37] V. Vyas, P. Pantel and E. Crestan. 2009. Helping Editors Choose Better Seed Sets for Entity Expansion. In *Proceedings of CIKM-09*.

[38] Y. Wong and H. T. Ng. 2007. One Class per Named Entity: Exploiting Unlabeled Text for Named Entity Recognition. In *Proc. of IJCAI-07*.

[39] D. Wu, W. S. Lee, N. Ye, and H. L. Chieu. 2010. Domain adaptive bootstrapping for named entity recognition. In *Proc. of EMNLP*.

[40] R. Yangarber, W. Lin and R. Grishman. 2002. Unsupervised Learning of Generalized Names. In *Proc. of COLING*.

View publication stats